Label Likelihood Maximisation: Adapting iris segmentation models using domain adaptation

Anton Mølbjerg Eskildsen
Department of Computer Science
IT University of Copenhagen
Copenhagen, Denmark
aesk@itu.dk

Dan Witzner Hansen
Associate professor
Department of Computer Science
IT University of Copenhagen
Copenhagen, Denmark
witzner@itu.dk

ABSTRACT

We propose to use unlabelled eye image data for domain adaptation of an iris segmentation network. Adaptation allows the model to be less reliant on its initial generality. This is beneficial due to the large variance exhibited by eye image data which makes training of robust models difficult. The method uses a label prior in conjunction with network predictions to produce pseudo-labels. These are used in place of ground-truth data to adapt a base model. A fully connected neural network performs the pixel-wise iris segmentation. The base model is trained on synthetic data and adapted to several existing datasets with real-world eye images. The adapted models improve the average pupil centre detection rates by 24% at a distance of 25 pixels. We argue that the proposed method, and domain adaptation in general, is an interesting direction for increasing robustness of eye feature detectors.

CCS CONCEPTS

• **Computing methodologies** → **Image processing**; **Transfer learning**; *Supervised learning by classification*; *Neural networks*; Online learning settings.

KEYWORDS

domain adaptation, deep learning, iris segmentation

ACM Reference Format:
Anton Mølbjerg Eskildsen and Dan Witzner Hansen. 2020. Label Likelihood Maximisation: Adapting iris segmentation models using domain adaptation. In *Symposium on Eye Tracking Research and Applications (ETRA '20 Full Papers), June 2–5, 2020, Stuttgart, Germany.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3379155.3391327

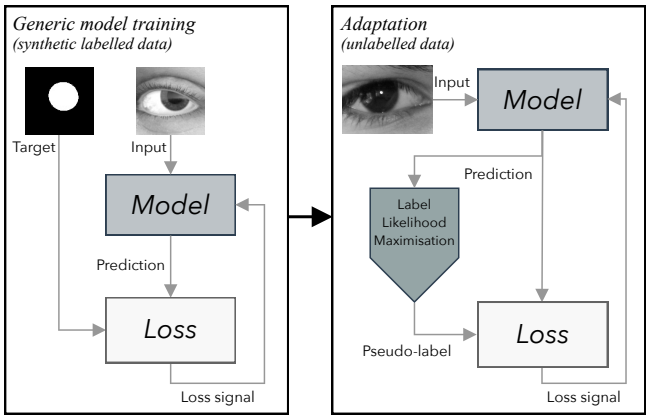

Figure 1: Overview of the method proposed in this paper. A generic model is trained using annotated synthetic data. The model is then used as a base for models adapted to specific circumstances, such as different capturing equipment or head-pose constraints. The creation of pseudo-labels enable adaptation without additional labelled data.

1 INTRODUCTION

This paper proposes a method for adapting an iris-segmentation network to specific datasets using a novel unsupervised domain adaptation technique.

Eye-tracking and eye detection methods based on machine learning techniques depend on the quantity and quality of the data used for training. Large and varied eye datasets exist and are used in state-of-the-art pupil detectors [Vera-Olmos et al. 2018; Wangwiwattana et al. 2018] and other models for eye tracking [Luo et al. 2019; Park et al. 2018; Zhang et al. 2017]. Despite their availability and size, eye-tracking data, especially in in-the-wild settings, suffer from biases due to the large variation in recording equipment, use settings, light conditions, and subject variation. A biased dataset is here defined as an eye image sample that is not independent and identically distributed with respect to the distribution of all possible eye images. The continued research in robust eye-tracking methods for in-the-wild settings is evidence that data bias has a concrete impact on the difficulty of training robust models. Dataset bias can be empirically observed in several available datasets on remote [Krafka et al. 2016; Zhang et al. 2017] and head-mounted [Fuhl et al. 2015, 2016; Tonsen et al. 2016; Wangwiwattana et al. 2018] eye image data. Synthetic eye images generated using physically accurate eye models facilitate the generation of larger and more consistent datasets without the need for manual data annotation. These datasets have been successfully used for gaze estimation [Park et al. 2018; Wood et al. 2016]. Image generators allow for datasets that are unbiased with regards to the rendering model.

Synthetic datasets are however still biased with regards to real eye images because they only approximate their appearance.

Bias is problematic because machine learning models depend directly on the training data. Using a model trained on data with a different sample distribution than the usage or test data results in performance degradation. This has been observed directly for object classification tasks [Torralba and Efros 2011]. This paper is motivated by the concept of *transduction* as defined by [Gammerman et al. 1998]. Transduction is the process of learning from biased training cases to biased test cases. This is opposed to *induction* which is the process of learning a general model for both training and test cases using just the training data. Applied to eye tracking, transduction makes it possible to make models more robust by learning the specific biases present in individual datasets.

Domain adaptation is a subfield of transfer learning concerned with applying transductive learning to machine learning models. This paper presents a novel domain adaptation technique that enables adaptation of an iris segmentation network to specific datasets using a label prior. In this context, priors refer to the distribution of labels before the knowledge of an input. In case of pixel-wise iris segmentation, the labels are binary images with pixels representing a Bernoulli distributed probability of that pixel belonging to the iris or not. We use the elliptical disk as a prior in the rest of this paper. Given a pre-trained model, the prior can be used to improve predictions when using the model on another dataset with a different bias. Specifically, we generate pseudo-labels by maximising the likelihood of a label given the model prediction under the specified prior, hence the name *label likelihood maximisation* (LLM). The pseudo-labels are used in on-line domain adaptation using the same supervised learning framework used to train the initial model. An overview of the method is shown in Figure 1.

In this paper, we present the theoretical background for the label likelihood maximisation technique as well as a concrete implementation for the task of iris segmentation. A fully convolutional neural network is used to produce a pixel-wise segmentation of the iris in eye images. The network is initially trained using synthetic data from [Wood et al. 2016] and then adapted to multiple real-world datasets using an implementation of LLM. The main contribution of this paper is the label likelihood maximisation method and its demonstration using iris segmentation. The implementation of LLM for iris segmentation and theoretical motivation for the effectiveness of using the generated pseudo-labels for domain adaptation is described in section 3. The experimental setup is presented in section 4 and results in section 5. Concluding remarks as well as a discussion of how LLM is generalisable to other eye tracking and eye detection tasks is presented in section 6.

2 RELATED WORK

The work presented in this paper is related to several fields in eye-tracking and eye-detection as well as general machine learning research.

Pupil detection methods. Algorithmic methods for pupil detection are the norm in head-mounted eye-tracking and have, until recently, shown superior performance compared to appearance-based methods. They typically involve edge detection and selection combined with statistical criteria for robust detection of the pupil centre or circumference. The detected pupil position is then used for gaze estimation using regression-based methods. ExCuSe and ElSE [Fuhl et al. 2015, 2016] both use a multi-stage approaches with fallbacks if candidates are not found initially while PuRE [Santini et al. 2018a] uses only one detection stage. While PuRE eclipses the detection rates of ExCuSE and ElSE, it suffers in certain situations when reflections or occlusions degrade the detected edges. Although algorithmic methods have achieved very high precision they are still bound by whatever approach has been chosen for detection. They typically suffer when extreme conditions of reflections and occlusions occur.

Machine learning approaches for pupil detection have currently surpassed the performance of algorithmic approaches. PupilNet [Wangwiwattana et al. 2018] and DeepEye [Vera-Olmos et al. 2018] both use the pupil centre. PupilNet predicts the probability of each pixel being the pupil centre and uses the expectation over a region as its centre candidate. DeepEye predicts a circular region of pixels surrounding the pupil centre and uses the centre of a predicted component as its candidate. The post-processing steps of these models imply constraints on outputs. They are, however, not used in the current implementations. CBF [Fuhl et al. 2018] uses weak pixel classifiers and assumptions on the geometry of the pupil to produce the highest detection rates of the examined pupil detection methods. It produces probabilities for a position being the pupil centre and uses the maximum as the final candidate.

Other methods used for eye tracking. Eye-tracking methods can be divided into appearance-based and shape/feature-based methods [Hansen and Ji 2010]. Appearance-based methods map an image directly to gaze while feature-based methods first estimate some arbitrary eye shape or feature metrics which are then used for gaze estimation. Pupil detection is, therefore, a feature-based method.

For remote eye tracking, the iris is a more appropriate target since its much larger radius makes it easier to detect accurately. It is used in [Park et al. 2018] together with eyelid landmark detection to construct a three-dimensional eye model for gaze estimation. Although landmark-estimation has shown promising accuracy, segmentation provides more information for subsequent steps as the whole boundary is estimated.

Generative models. Several generative methods have been proposed for decreasing the reliance on manually collected and annotated data. Directly generating eye images using a physically accurate eye model has been used for competitive gaze estimation in unconstrained [Wood et al. 2016] and VR [Kim et al. 2019] settings. The former method has been used in the aforementioned landmark-based gaze estimation system. Generated data allows precise control over the generated distribution of images. The disadvantage of using exclusively generated data is the inevitable disparity between generated and real data. In NVGaze [Kim et al. 2019], the models are trained on both generated and real data which counteracts the disparity issue. However, this approach does not alleviate the need for large annotated datasets. A more direct generative approach is used in [Eivazi et al. 2019] where a feature-based method (PuRE) is used to create a dataset of easy samples. The dataset is subsequently augmented to include reflections, fake glints, and fake pupils. The method eclipses DeepEye in performance despite not using manually annotated data.

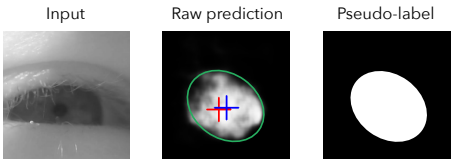

Figure 2: Example of iris segmentation output (middle) and pseudo-label produced by LLM (right). The pseudo-label has been drawn onto the original prediction to ease comparison. The red cross indicates the true pupil position and the blue cross indicates the centre of the pseudo-label.

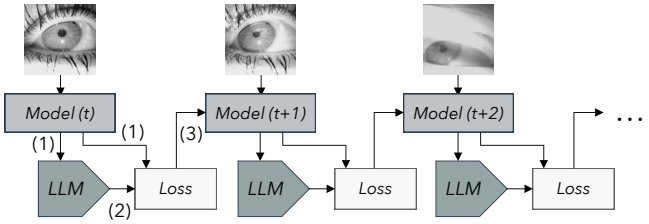

Figure 3: Shows how LLM is applied in practice. The base model is adapted continuously throughout training.

3 METHOD

This section describes the details of the label likelihood maximisation method and iris segmentation network as well as the theoretical motivation for using priors to generate pseudo-labels.

We define iris segmentation as the task of predicting whether each pixel in an input image belongs to the iris or not. This task is performed by a predictive function $f_\theta(x) : \mathcal{X} \rightarrow \mathcal{Y}$ which in this case is a neural network. Here, $\mathcal{X}$ is the input feature space and $\mathcal{Y}$ is the label space. Inputs are 224×224 grey-scale images and labels are 224×224 binary segmentation maps. Each output of $f_\theta(x) \in \mathcal{X} \rightarrow \mathcal{Y}$ represents a Bernoulli distributed probability of that pixel belonging to the iris. Let $\Phi : \mathbb{F} \rightarrow \mathcal{Y}$ be a function from parameter space $\mathbb{F}$ to label space $\mathcal{Y}$. For iris segmentation, the $\mathbb{F}$ is the five parameters of an elliptical disk and Φ maps from parameters to the binary label space. A label y is valid if $y = \Phi(h)$ for some $h \in \mathbb{F}$. Thus Φ acts as a prior on the geometry of valid iris segmentation outputs.

In this paper, Φ is used in post-processing of network predictions as well as in pseudo-label generation. During post-processing, the segmentation output is processed to find connected regions consisting of pixels belonging to the iris. An ellipse is fitted to the circumference points of each region. Each ellipse candidate is scored according to the following confidence measure:

$$\frac{\|\hat{y} - \widetilde{y}\|}{\|\widetilde{y}\| n}, \tag{1}$$

where $\|\|$ denotes the L1-norm, $\widetilde{y}$ is the pixel-wise segmentation, $\hat{y} = \Phi(h)$ is the fitted ellipse with parameters h, and n is the number of components. n is included for penalising multi-component predictions during domain adaptation as explained in the theoretical motivation (subsection 3.1). The ellipse with the highest confidence is selected as the final candidate. An example ellipse fitting including drawing into a binary image is shown in Figure 2.

Φ is used for unsupervised domain adaptation by generating pseudo-labels using the method just described. An overview is shown in Figure 3. The pseudo-label is used in place of a ground-truth annotation to further optimise the base model using gradient descent. The updated model is used to generate the next pseudo-label.

3.1 Model details

The pseudo-label generation method proposed for iris segmentation is theoretically motivated in this section using a probabilistic framework. The name label likelihood maximisation stems from the fact that the method for finding the pseudo-label is equivalent to maximising the likelihood of the pseudo-label given a specific prediction. We make some assumptions under which it is possible to argue that training using the generated pseudo-labels leads to better results. The validity of these assumptions in practice is discussed in the results section of the paper. The probabilistic interpretation also lends itself to generalisation to other tasks. Possibilities will be considered in the discussion.

Let a machine learning problem be defined by a joint probability distribution $p(X, Y)$, where $X \in \mathcal{X}, Y \in \mathcal{Y}$ are random variables, representing the input (eye images) and output (segmentations) respectively. Unlabelled samples $\{x_i\}$ are drawn from $p(X)$ and labelled samples $\{x_i, y_i\}$ are drawn from $p(X, Y)$. The goal of machine learning is to find a predictive function $f_\theta(x)$ that approximates the conditional distribution $p(Y|X)$. When the training and test data are biased, they are not drawn directly from $p(X, Y)$ but instead from biased distributions $p(X, Y)^s$ and $p(X, Y)^t$ for training and test data respectively. If samples $\{x_i, y_i\}$ are drawn from $p(X, Y)^s$ and used in training, the resulting model $f_\theta(x)$ approximates a biased conditional distribution $p(Y|X)^s$. Predictions are therefore less accurate for samples drawn from the biased test distribution.

Given model $f_\theta(x)$ trained on data from $p(X, Y)^s$, the goal of domain adaptation is to modify $f_\theta(x)$ to accurately estimate the target conditional distribution $p(Y|X)^t$ for samples x_i^t drawn from the marginal distribution of the test data $p(X)^t$. LLM models the prediction error when using $f_\theta(x)$ on biased samples as a random variable $\widetilde{Y}$. Specifically, $\widetilde{Y}$ is defined as the distribution of predictions for samples with the same ground-truth result. Multiple eye images with the same iris segmentation exist because the space of possible eye images is much larger than the space of iris segmentations. For example, differences in the appearance of the eye does not affect the position of the iris. The definition of $\widetilde{Y}$ is thus physically realistic. We express the distribution of $\widetilde{Y}$ as

$$\widetilde{Y} \sim p(\widetilde{y}|y) = p(y|X) + \epsilon \tag{2}$$

where $p(\widetilde{y}|y)$ is the conditional distribution of predictions given a ground-truth annotation, $p(y|X)$ is constant by definition and ϵ is a random variable with unknown distribution.

Constraining labels. This section describes how the generation of pseudo-labels maximises the likelihood of predictions when using the ellipse model as a prior on the distribution of labels. A pseudo-label is generated by finding the most likely point in the range of Φ given a model prediction $\widetilde{y}$. The optimal pseudo-label is found by maximising $p(\widetilde{y}|y)$ substituting true labels for possible

pseudo-labels:

$$\hat{y} = \arg\max_{\Phi(h)} p(\widetilde{y}|\Phi(h)), \tag{3}$$

Assume $p(\widetilde{y}|\Phi(h))$ is unimodal with its mode at $\widetilde{y} = \Phi(h)$. This makes the solution to Equation 3 equivalent to minimising the euclidean distance between $\widetilde{y}$ and $\hat{y}$ subject to $\hat{y} = \Phi(h)$ for $h \in F$. The formulation can be adapted to other distance measures as long as the above assumption is met. This is the case for ellipse fitting which minimises the distance between the ellipse circumference to detected contour points. We have thus shown that the elliptical disk produced from the post-processing step in the presented iris segmentation network is the one that maximises the likelihood of the prediction with the elliptical disk used as prior.

If the prior is not used, the definition of $p(\widetilde{y}|\Phi(h))$ ensures that $\hat{y} = \widetilde{y}$ which results in a loss of 0 when using $\hat{y}$ for training. The prior is therefore essential to the method.

Error reduction and confidence. The probabilistic interpretation allows analysis of how the pseudo-labels $\hat{y}$ compare to the unchanged network predictions $\widetilde{y}$. This is important to understand how the pseudo-labels impact network performance when using them for domain adaptation.

Assume that the range of Φ is a linear, k-dimensional subspace H of the n-dimensional label space $\mathcal{Y}$ and H and $\mathcal{Y}$ are orthonormal. For the iris segmentation task, $k = 5$ and $n = 224^2$. Under this assumption, LLM can be expressed as the projection of $\widetilde{Y}$ onto H, i.e. $\hat{Y} = H\widetilde{Y}$. The expected ratio of the distance between the true label and prediction and pseudo-label is then

$$\frac{\mathbb{E}[\|\widetilde{Y} - y\|_2]}{\mathbb{E}[\|H\widetilde{Y} - y\|_2]} = \frac{\sqrt{\sum_{i=1}^{n} Var(\widetilde{Y}_i - y)_i}}{\sqrt{\sum_{i=1}^{k} Var(\widetilde{Y}_i - y_i)}},$$

where i is used to denote a specific element. This is significant because $k << n$ for the presented iris segmentation problem. To demonstrate the ratio, if $Var(\widetilde{Y})_1 = \cdots = Var(\widetilde{Y})_n$, the distance ratio is $\sqrt{n/k}$. LLM thus acts as a form of variance reduction.

Φ as used in the iris segmentation task is not linear. All linear functions must have the following property

$$f(x + y) = f(x) + f(y).$$

For Φ, the output which belongs to the disk is 1 and everything else is 0. Therefore, adding two overlapping ellipses results in at least one pixel with the value 2 which is not valid. Φ is therefore not linear. The assumption above is therefore changed to require only approximate local linearity, i.e. $\Phi(x + h) - \Phi(x) - \Phi(h) < \epsilon$, for a small arbitrary value ϵ. To incorporate this into the training process, a confidence measure is added to measure locality. For iris segmentation, the same measure used for candidate selection, Equation 1 is used.

Training. A final assumption has to be made for LLM based domain adaptation to converge. For a given set of training pairs $x_1, \ldots, x_n$, we assume the predictions $\widetilde{Y}$ are distributed such that

$$\mathbb{E}[p(\widetilde{Y} = y)] > 0.5. \tag{4}$$

This means that the majority of predictions $\widetilde{y}$ are correct. Pseudo-labels have the same property since $\hat{y}$ are assumed to be better estimates than $\widetilde{y}$. During training, any $\hat{y} \neq y$ therefore contribute less to the total error than the correct pseudo-labels where $\hat{y} = y$.

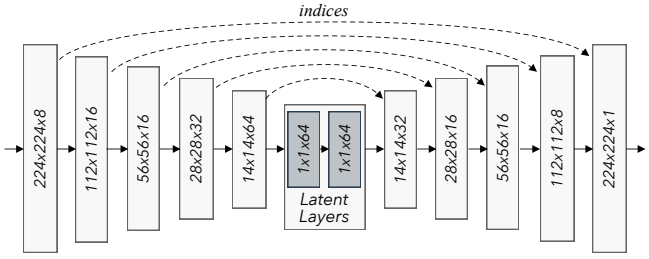

Figure 4: Overview of the segmentation network architecture. The dimensions defined in parentheses denote the dimensions of the feature maps. The decoder blocks use indices saved from the encoder blocks for upscaling using max-unpooling. The latent layers use convolutions but are still equivalent to regular dense layers do to their size.

Since LLM acts as a variance reduction technique it increases the probability of $p(\widetilde{Y} = y)$ (when Equation 4 is true). Otherwise, additional training causes overfitting. This is discussed in section 5.

The loss function used for domain adaptation is

$$J(\theta) = \mathbb{E}[\alpha \log p(\hat{y}|x)], \tag{5}$$

where $p(\hat{y}|x)$ is the Bernoulli density function and α denotes the confidence measure. Sample weighing is possible in most gradient descent based optimisation frameworks and the method therefore works with very little adaptation of the typical supervised learning routine.

To summarise, the convergence and effectiveness of using LLM for domain adaptation depends on the following assumptions:

(A) $p(\widetilde{y}|\Phi(h))$ is uni-modal and has maximum at $\widetilde{y} = \Phi(h)$.
(B) Φ is approximately linear locally, i.e. $\Phi(x+h)-\Phi(x)-\Phi(h) < \epsilon$, for an arbitrary value ϵ.
(C) $\widetilde{Y}$ given the ground-truth label y is distributed such that $\mathbb{E}[p(\widetilde{Y} = y)] > 0.5$.

3.2 Segmentation network implementation

This section describes details of the network used to perform pixel-wise iris segmentation. Figure 4 shows the model used for iris segmentation. It uses a fully convolutional encoder/decoder network similar to DeconvNet, a network created for general semantic segmentation [Noh et al. 2015]. The encoder creates a flat latent representation of the input. The decoder uses the latent representation to reconstruct the iris segmentation map. The encoder is a regular convolutional neural network (CNN) with a series of convolutional and max-pooling layers. Due to the destruction of spatial information caused by max-pooling, the latent representation does not contain sufficient spatial information for accurate reconstruction [Zeiler et al. 2011]. The decoder, therefore, uses max-unpooling for upsampling, which reuses indices used in corresponding max-pooling downsampling layers in the encoder. This results in more spatial information being retained in the final segmentation output as demonstrated in [Zeiler et al. 2011].

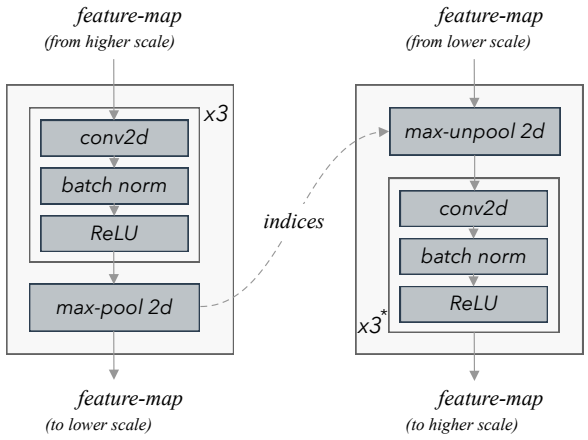

Figure 5: Details of an encoder block (left) and its corresponding decoder block (right). Indices from the max-pooling operation in the encoder block is used for max-unpooling in the decoder block of equal scale, i.e. the lines flow as shown in Figure 4.

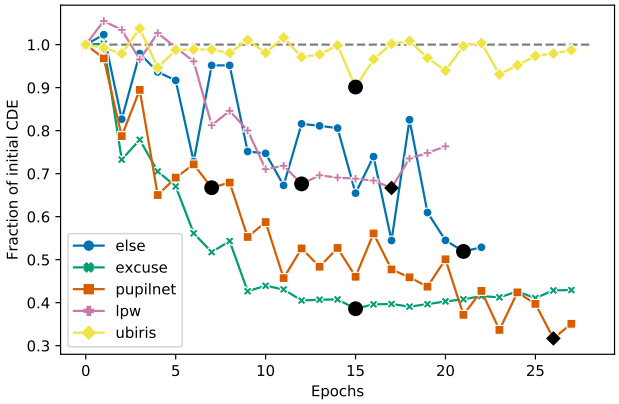

Figure 6: Graph showing the distance error as a function of training epochs. The error is shown as a fraction of the initial error to normalise the scale. The black circles indicate the final model used for analysis. The black diamonds indicate additional models used for the analysis of overfitting behaviour.

Scale blocks. The encoder and decoder each contain 5 scale blocks. Each block in the encoder contains 3 convolutional layers with batch normalisation and ReLU activations followed by a max-pooling layer (shown in Figure 5). The indices used in max-pooling are used for the max-unpooling operation in the corresponding decoder block. Each decoder block consists of an upscaling layer followed by three convolutional layers, each with batch normalisation and ReLU activations (except the last layer which uses a sigmoid activation to produce binary outputs). Regular convolutions are used instead of transposed convolutions as the latter tends to produce chequerboard shaped artefacts [Odena et al. 2016]. The flat latent layers are created using convolutions with kernel size equal to the feature maps of the last encoder block. The initial upsampling as shown in Figure 4 is achieved using a deconvolution layer. A ReLU activation follows each convolutional layer in the latent block.

4 EXPERIMENTS

This section describes the experiments performed with the iris segmentation network and LLM. A base network is trained using synthetic data from UnityEyes [Wood et al. 2016]. The base model is adapted using LLM to a number of real-world datasets, resulting in one adapted model for each dataset. The synthetic data is split in a ratio of 0.6/0.2/0.2 for training, validation, and testing. Because the domain adaptation step is unsupervised, the real-world datasets have not been split. In other words, the images used to adapt the generic model to a specific dataset are the same used in testing. For each dataset, training is continuous and uses all the included images.

The synthetic training dataset contains 1, 093, 496 images with ±20° variation in camera rotation around the eyeball centre and ±30° variation of eyeball rotation with uniformly random distributions. The generator creates varied images with respect to iris-colour, ethnicity, face shape, and reflections.

The real-world datasets comprise a number of large-scale head-mounted eye image datasets with pupil center annotations [Fuhl et al. 2015, 2016; Tonsen et al. 2016; Wangwiwattana et al. 2018] as well as a smaller dataset [Proenca et al. 2010] of 2250 images with iris segmentation annotations. In total, 267, 796 images. The use of datasets with pupil-centre annotations is motivated by the fact that few datasets exist with annotated iris-segmentations and the ones that do, including Ubiris, are largely used for the purpose of testing iris-recognition algorithms. They are thus not expected to be as challenging as the included pupil-centre datasets which have been created explicitly to represent challenging real-life eye-tracking situations. We are aware that the iris and pupil centre do not necessarily coincide but since the main purpose of these experiments is to demonstrate the potential of LLM, we argue that the application and analysis of LLM's performance in real-world scenarios are of higher importance than precise comparability to methods that directly detect pupil centres.

Adam is used for model parameter optimisation [Kingma and Ba 2014] and binary cross-entropy used as the loss function. The base model used a learning rate of 1×10^{-4} and weight decay of 1×10^{-5}. The specialisation used 1×10^{-6} for the learning rate and 1×10^{-7} for weight decay. Base model training progressed with early stopping checks on the F1-score (the harmonic mean of the precision and recall [Goodfellow et al. 2016]) and patience of 2 epochs. LLM adaptation progressed for a variable number of epochs for each dataset. In each case, the mean pupil centre distance error was used to choose the optimal model. The relative decrease in error compared to the base model is shown for each specialisation run in Figure 6.

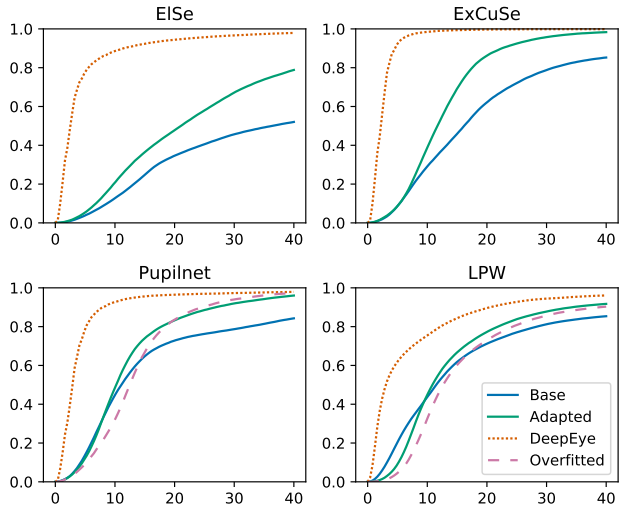

Figure 7: Detection rate as a function of distance. Each plot shows a separate dataset. The *adapted* model for each graph represents the model adapted to that dataset using LLM. Results from DeepEye [Vera-Olmos et al. 2018] are shown for comparison.

All synthetic data have an original size of 640×480 pixels and is cropped to 320×320 pixels (centred) for training and testing. All data captured with real cameras were resized to 384×288 pixels before being cropped from the centre to 288×288 pixels. All data was finally resized to 224×224 pixels for input into the network. Labels for the rendered data were generated by fitting the landmark points produced by UnityEyes to an ellipse and drawing it onto the label image. Data augmentation was used for all training with random translations of ±10% of the input width and height, random rotations of ±15°, and random scaling in the interval $[0.95, 1.05]$ times the original size for the generic model and $[0.95, 1.15]$ for domain adaptation.

5 RESULTS

The results are based on ellipses fitted to the segmentation output using the described post-processing method. For the pupil centre annotated data, the detected iris centre is used as a pupil centre approximation. For evaluating segmentation performance using IOU (intersection over union), the pseudo-label $\hat{y}$ is used as the final prediction. Any improvement caused by adapting the models can thus not be explained by the simple addition of a geometric constraint on individual outputs. We count rejected detections for both the iris segmentation network as well as the comparisons.

Figure 7 shows the pupil detection rate as a function of distance error for the *base* and *adapted* models as well as for the DeepEye model [Vera-Olmos et al. 2018]. Note that the detection rates have been scale-corrected for the difference in scale between segmentation output and source image size. Specific detection rates at distances of 5 and 25 pixels are shown in Table 3 for the iris segmentation models as well as several state-of-the-art and well-known

Table 1: Change in percent of several metrics between the base model and each adapted model (specified by dataset).

	Change in percent			
Dataset	Detection rate	Mean	Std	Median
ElSE	42.99	-50.36	-40.14	-42.08
ExCuSe	28.25	-55.75	-72.75	-27.59
Pupilnet	16.15	-49.21	-64.39	-8.03
LPW	8.81	-25.93	-22.23	-5.57
Ubiris	0.18	-2.85	-8.59	-1.21

Table 2: Number of rejected samples for each model/dataset combination.

	Model					
Dataset	Base	Adapted	ElSE	ExCuSe	PuRE	DeepEye
ElSE	2225	326	366	4483	121	19
ExCuSe	964	58	795	1651	34	8
Pupilnet	708	1305	1152	3420	138	51
LPW	2960	4948	305	10850	409	560
Ubiris	0	0	0	0	0	1

pupil detection algorithms. The 25 pixel threshold is included because of the disparity between pupil and iris centre making iris segmentation an inherently imprecise method for estimating the pupil centre directly. It is important to note that our primary interest is to observe any improvement caused by adaptation. Table 1 shows the relative improvement of the adapted models compared to the base model. It includes the detection rate at 25 pixels as well as the mean, standard deviation, and median of the detection distance. The average improvement in the detection rate for the pupil centre datasets is 24.05%. Notice that samples discarded by the network are not counted. The number of rejected samples for each model is shown in Table 2.

Effect of LLM on model performance. The detection rate is improved on all tested datasets. The reduction in the mean and standard deviation of detection distance is a clear indicator that LLM improves both the reliability and precision of the base network. Because the networks use the same post-processing operations which are identical to the LLM pseudo-label generation method, these improvements cannot be explained by a single step pseudo-label generation. Instead, the models have learned to better infer results by training only using the pseudo-labels.

Because the rejected samples have been removed from the detection distance metrics, their impact must also be considered. For the ExCuSE and ElSE datasets, adaptation greatly decreases the number of rejected samples while they are increased for the Pupilnet and LPW datasets. This is related to overfitting. When the models are trained for enough iterations, the models for PupilNet and LPW end up with a decreased number of rejections. These models are also graphed in Figure 7 as *overfitted*. A lower number of rejections is thus not necessarily an indicator of robustness. On the contrary, a rejection makes it possible to know when a detection has failed and adjust further processing of the result accordingly.

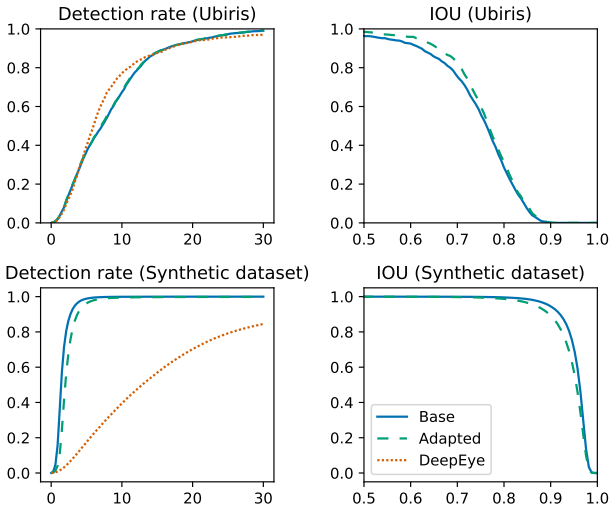

Figure 8: The two left plots show the detection rate as a function of distance for the Ubiris and synthetic test datasets. Here, results from DeepEye[Vera-Olmos et al. 2018] are included for comparison. The right plots show intersection over union (IOU) for the iris segmentation models only.

The domain adaptation seems to degrade the detection rate at low distances for some datasets after many training iterations. This is an indication of overfitting. Since only the geometry of a prediction is considered when using LLM, the prediction itself acts as an implicit ground truth on which the pseudo-label depends. When overfitting, the network is trained on pseudo-labels which, despite being geometrically correct, are not correct with respect to the input image. This causes the network to produce increasingly skewed predictions despite being more and more geometrically probable. In other words, as training progresses, Equation 4 becomes less and less likely while the network becomes increasingly confident in its predictions. This also explains why the overfitted models have a lower number of rejections. This shows the inherent weakness of only considering the geometry of predictions when creating pseudo-labels.

Segmentation performance. Figure 8 shows the results for the segmentation tests on the Ubiris and synthetic test datasets. The detection rates are included as functions of iris distance error as well as the intersection over union (IOU). The results for the synthetic dataset are based on the same models used for the Ubiris dataset. It is included to evaluate the degradation of performance in the original training domain after adaptation.

The adapted model only increases the detection rate by 0.19% at 25 pixels and the mean IOU by 0.56%. This may partly be caused by the small size of the dataset compared to the ones used for pupil detection as well as the relatively high performance of the base model. This results in small distances, i.e. $\mathbb{E}[\|\widetilde{Y}-y\|_2]$ is small, when creating pseudo-labels leading to smaller changes in the network. Additionally, the low number of samples increases the variance of Equation 4 (central limit theorem). It is thus more likely that the assumption doesn't hold and that a majority of pseudo-labels are wrong.

Comparison to other methods. We have included the algorithms/models ElSE, ExCuSE, PuRE, and DeepEye for comparison as they have readily available implementations and represent well-known and state-of-the-art pupil detection algorithms. DeepEye is the only one that is based on neural networks and is trained directly using pupil centre annotations. The network target is a disk with a fixed radius and centre at the annotated pupil location. The other methods use various techniques that are based on detection and analysis of edges and pixel intensities. They are static, algorithmic approaches.

The strength of deep learning-based methods is that they learn directly from data. This makes it easier to make models that are robust to situations that make edge detection or intensity-based approaches hard to use. This is shown in the detection rates at 25 pixels, where our models mostly outperform the algorithmic approaches. At 5 pixels, DeepEye has the highest performance in all but the LPW dataset, where PuRE outperforms it. It is important to note that better methods have since been presented using deep learning [Eivazi et al. 2019] and by continuously tracking the pupil over time [Santini et al. 2018b].

DeepEye is most directly comparable to our iris segmentation network. Even when evaluated on the Ubiris dataset where the iris centre is annotated, DeepEye still has a higher detection rate than our adapted models at 5 pixels. It is important to note, however, that while DeepEye was trained directly on real eye data, our base model was trained on synthetic data. The results from the synthetic test set show that our models are indeed capable of very precise centre detection as well as precise segmentation.

Table 3: Detection rate at a distances of 5 and 25 pixels.

	5 pixels						25 pixels					
Dataset	Base	Adapated	ElSE	ExCuSe	PuRE	DeepEye	Base	Adapated	ElSE	ExCuSe	PuRE	DeepEye
ElSE	0.04	0.05	0.46	0.33	0.57	0.78	0.39	0.58	0.60	0.50	0.71	0.96
ExCuSe	0.08	0.09	0.72	0.52	0.79	0.92	0.70	0.92	0.82	0.69	0.89	1.00
Pupilnet	0.14	0.09	0.50	0.28	0.60	0.79	0.75	0.90	0.64	0.47	0.72	0.97
LPW	0.20	0.05	0.76	0.47	0.82	0.62	0.75	0.81	0.87	0.79	0.89	0.92
Ubiris	0.37	0.37	0.22	0.07	0.22	0.40	0.97	0.97	0.59	0.56	0.59	0.96
Synthetic	0.99	0.96				0.17	1.00	1.00				0.79

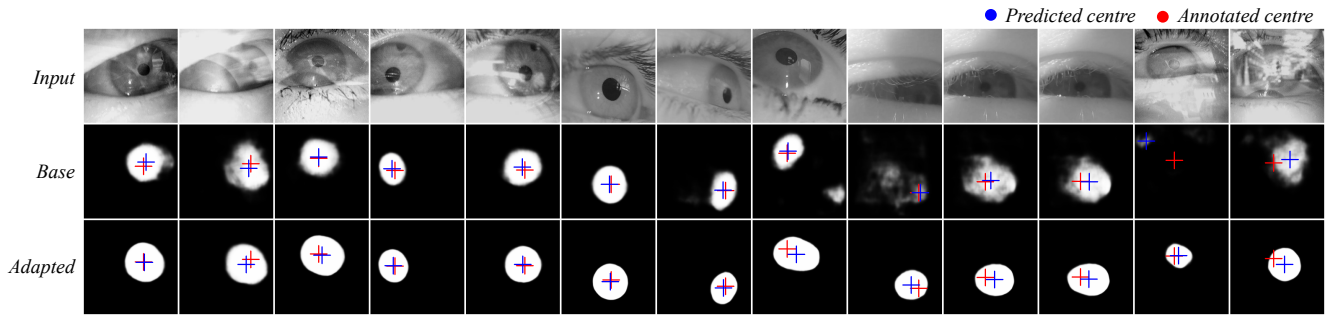

Figure 9: Example predictions from the real datasets used for testing. The first row contains the cropped and resized images, the second is the base model predictions, and the third is the *adapted* model predictions. The blue crosses indicate the predicted centre and the red crosses indicate the true pupil centre.

Inference speed. On a midrange GPU (NVidia Geforce GTX 970), base inference takes 17.8*ms* per image, equivalent to 56.2 frames per second (FPS). On a four-core CPU (Intel I7 6700k, 4GHz base clock speed), inference takes 300*ms* per image (3 FPS). These times were recorded using images from the ExCuSE dataset and include time to load, infer segmentation, and perform post-processing.

We additionally tested speeds when using batches of images. While this increases the inference delay, batch processing makes inference much faster. Using the same GPU, an inference of 7.65*ms* per image (131 FPS) was achieved using batches of four images. On the CPU, batches of four resulted in an inference speed of 150*ms* per image (6 FPS).

Limitations. LLM is limited by how strongly the assumptions made on the distribution of predictions hold in given situations. Since no sources directly inform the model of correct labels during training, overfitting to the prior model can easily occur. This was observed in some of the adapted models when trained for enough iterations. Further experimentation is necessary to accurately understand the practical limitations of relying solely on geometric constraints for adaptation. An interesting direction for future work is to introduce weak detectors for features that can be used as proxies for the real feature, e.g. using the pupil centre to guide the creation of a pseudo-label for iris segmentation.

In its current form, LLM may be difficult to adapt to more complex problems that do not have an easy geometric solution or combines multiple geometric models. Creating a general implementation of LLM that is differentiable will allow gradient descent optimisers to estimate pseudo-labels for geometric models of arbitrary complexity.

6 DISCUSSION

This paper proposed label likelihood maximisation, a generally applicable method for adapting an iris segmentation network using unsupervised domain adaptation. The method results in significant improvements on several datasets. Based on these findings, we argue that the method and framework as a whole is an interesting direction for future research in eye-tracking research as it enables more general applications of created models without relying on any specific data source.

LLM can easily be extended to other machine learning problems for which it is possible to define a label prior as a function Φ. Since the expected distance reduction is dependent on the difference in the number of dimensions of $\mathbb{F}$ and $\mathcal{Y}$, tasks that involve a large reduction in the number of dimensions from model output to final prediction are expected to work best. This includes the pupil-detection models DeepEye, Pupilnet, and the landmark detection model presented in [Park et al. 2018]. The label prior function Φ can be defined for these approaches using their post-processing steps. They are therefore obvious candidates for testing the general applicability of LLM.

REFERENCES

Shaharam Eivazi, Thiago Santini, Alireza Keshavarzi, Thomas Kübler, and Andrea Mazzei. 2019. Improving Real-Time CNN-Based Pupil Detection through Domain-Specific Data Augmentation. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, Colorado) *(ETRA '19)*. Association for Computing Machinery, New York, NY, USA, Article 40, 6 pages. https://doi.org/10.1145/3314111.3319914

Wolfgang Fuhl, David Geisler, Thiago Santini, Tobias Appel, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2018. CBF: Circular Binary Features for Robust and Real-Time Pupil Center Detection. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) *(ETRA '18)*. Association for Computing Machinery, New York, NY, USA, Article 8, 6 pages. https://doi.org/10.1145/3204493.3204559

Wolfgang Fuhl, Thomas Kübler, Katrin Sippel, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2015. ExCuSe: Robust Pupil Detection in Real-World Scenarios. In *Computer Analysis of Images and Patterns*, George Azzopardi and Nicolai Petkov (Eds.). Springer International Publishing, Cham, 39–51.

Wolfgang Fuhl, Thiago C. Santini, Thomas Kübler, and Enkelejda Kasneci. 2016. ElSe. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16*. ACM Press. https://doi.org/10.1145/2857491.2857505

A. Gammerman, V. Vovk, and V. Vapnik. 1998. Learning by Transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (Madison, Wisconsin) *(UAI'98)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 148–155.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

D. W. Hansen and Q. Ji. 2010. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 3 (March 2010), 478–500. https://doi.org/10.1109/TPAMI.2009.30

Joohwan Kim, Michael Stengel, Alexander Majercik, Shalini De Mello, David Dunn, Samuli Laine, Morgan McGuire, and David Luebke. 2019. NVGaze: An Anatomically-Informed Dataset for Low-Latency, Near-Eye Gaze Estimation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) *(CHI '19)*. ACM, New York, NY, USA, 10. https://doi.org/10.1145/3290605.3300780

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. 2016. Eye Tracking for Everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2176–2184. https://doi.org/10.1109/CVPR.

2016.239

M. Luo, X. Liu, and W. Huang. 2019. Gaze Estimation Based on Neural Network. In *2019 IEEE 2nd International Conference on Electronic Information and Communication Technology (ICEICT)*. 590–594.

H. Noh, S. Hong, and B. Han. 2015. Learning Deconvolution Network for Semantic Segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 1520–1528. https://doi.org/10.1109/ICCV.2015.178

Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and Checkerboard Artifacts. *Distill* (2016). https://doi.org/10.23915/distill.00003

Seonwook Park, Xucong Zhang, Andreas Bulling, and Otmar Hilliges. 2018. Learning to Find Eye Region Landmarks for Remote Gaze Estimation in Unconstrained Settings. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) *(ETRA '18)*. Association for Computing Machinery, New York, NY, USA, Article 21, 10 pages. https://doi.org/10.1145/3204493.3204545

H. Proenca, S. Filipe, R. Santos, J. Oliveira, and L. A. Alexandre. 2010. The UBIRIS.v2: A Database of Visible Wavelength Iris Images Captured On-the-Move and At-a-Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 8 (Aug 2010), 1529–1535. https://doi.org/10.1109/TPAMI.2009.66

Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. 2018a. PuRe: Robust pupil detection for real-time pervasive eye tracking. *Computer Vision and Image Understanding* 170 (2018), 40 – 50. https://doi.org/10.1016/j.cviu.2018.02.002

Thiago Santini, Wolfgang Fuhl, and Enkelejda Kasneci. 2018b. PuReST: Robust Pupil Tracking for Real-Time Pervasive Eye Tracking. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) *(ETRA '18)*. Association for Computing Machinery, New York, NY, USA, Article 61, 5 pages. https://doi.org/10.1145/3204493.3204578

Marc Tonsen, Xucong Zhang, Yusuke Sugano, and Andreas Bulling. 2016. Labelled pupils in the wild. *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16* (2016). https://doi.org/10.1145/2857491.2857520

A. Torralba and A. A. Efros. 2011. Unbiased look at dataset bias. In *CVPR 2011*. 1521–1528. https://doi.org/10.1109/CVPR.2011.5995347

F. Vera-Olmos, E. Pardo, Helena Melero, and N. Malpica. 2018. DeepEye: Deep convolutional network for pupil detection in real environments. *Integrated Computer-Aided Engineering* 26 (08 2018), 1–11. https://doi.org/10.3233/ICA-180584

Chatchai Wangwiwattana, Xinyi Ding, and Eric C. Larson. 2018. PupilNet, Measuring Task Evoked Pupillary Response Using Commodity RGB Tablet Cameras: Comparison to Mobile, Infrared Gaze Trackers for Inferring Cognitive Load. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 171 (Jan. 2018), 26 pages. https://doi.org/10.1145/3161164

Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. 2016. Learning an Appearance-Based Gaze Estimator from One Million Synthesised Images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications* (Charleston, South Carolina) *(ETRA '16)*. Association for Computing Machinery, New York, NY, USA, 131–138. https://doi.org/10.1145/2857491.2857492

M. D. Zeiler, G. W. Taylor, and R. Fergus. 2011. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*. 2018–2025. https://doi.org/10.1109/ICCV.2011.6126474

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (11 2017). https://doi.org/10.1109/TPAMI.2017.2778103